

WEB- & N-TIER TECHNOLOGIES

Eine Einführung in Web-Technologie und verteilte Applikationen

Autor: Walter Rothlin (24.5.21)

Inhalt

- http-Protokoll (request und response), Session, Stateless / Statefull → Analyse von Google.com mit Google Chrome Entwicklertools
- HTML-5 mit Form-Elementen / CSS → Design und Entwickeln einer Homepage nativ / Using bootstrap
- Request- / Response Web-Client entwickeln → Formhandler Application client side entwickeln
- Java-Script / Events and Event-Handlers / DOM → Interaktive Page mit JS entwickeln
- REST-Service (Open Weather, search, maps, QR-Code..) API verstehen und requests vom Browser her absetzen und Response analysieren → AJAX applikation client side entwickeln
- REST Requests von Python aus absetzen und response in Python auswerten
- Serverseitige Programme → CGI, WSGI, JS-Node, Flask, JEE, PHP

Von der Ablaufsteuerung zu Service orientiert

```
10 INPUT "Geben Sie bitte Ihren Namen ein"; A$
20 PRINT "Guten Tag, "; A$
30 INPUT "Wie viele Sterne möchten
35 S$ = ""
40 FOR I = 1 TO S
50 S$ = S$ + "*"
55 NEXT I
60 PRINT S$
70 INPUT "Möchten Sie noch mehr St
80 IF LEN(Q$) = 0 THEN GOTO 70
90 L$ = LEFT$(Q$, 1)
100 IF (L$ = "J") OR (L$ = "j") THEN GOTO 30
110 PRINT "Auf Wiedersehen";
120 FOR I = 1 TO 200
130 PRINT A$; " ";
140 NEXT I
150 PRINT
```

Sequenzen, if-goto

- Each application starts from scratch
- Maintenance intensive
- Migration tests
- Reuse via cut / past
- Predefined functions as part of the language

```

# -----
# Name: umrechnungen.py
#
# Description: Rechnet verschiedene physikalische Grössen um.
#
# Autor: Walter Rothlin
#
# History:
# 26-Sep-2017   Walter Rothlin   Initial Version
# -----

doLoop = True
while doLoop:
    print(" Umrechnungen")
    print(" =====")
    print(" 1: Grad in Bogenmass")    # rad = grad*pi/180
    print(" 2: Bogenmass in Grad") # grad = rad*180/pi
    print()
    print(" 3: Fahrenheit in Celsius") #32F -> 0°C    100F -> 38.8°C    °C = (°F - 32) / 1.8
    print(" 4: Celsius in Fahrenheit") #32F -> 0°C    100F -> 38.8°C    °F = (°C * 1.8) - 32
    print()
    print(" 0: Schluss")

    antwort = input("\n Wähle:")
    if (antwort == "1"):
        gradValue=float(input("Grad:"))
        print(gradValue, "Grad ==> ", gradValue*3.141592/180, "Rad")

    if (antwort == "2"):
        radValue=float(input("Rad:"))
        print(gradValue, "Rad ==> ", radValue*180/3.141592, "Grad")

    if (antwort == "3"):
        pass

    if (antwort == "4"):
        pass

    if (antwort == "0"):
        doLoop = False

print("Ende....Done")

```

Loops and If-Then-Else
+ Better Maintenance

```

# -----
def grad2Rad(grad):
    return 3.1415*grad/180

def rad2Grad(rad):
    return 180*rad/3.1415

def fahrenheit2Celsius(fahrenheit):
    return (fahrenheit-32)/1.8

def celsius2Fahrenheit(celsius):
    return (celsius*1.8)+32

doLoop = True
while doLoop:
    VT52_cls_home()
    print("  Umrechnungen")
    print("  =====")
    print("  1: Grad in Bogenmass")    # rad = grad*pi/180
    print("  2: Bogenmass in Grad")  # grad = rad*180/pi
    print()
    print("  3: Fahrenheit in Celsius")  #32F -> 0°C    100F -> 38.8°C    °C = (°F - 32) / 1.8
    print("  4: Celsius in Fahrenheit")  #32F -> 0°C    100F -> 38.8°C    °F = (°C * 1.8) - 32
    print()
    print("  0: Schluss")

    antwort = input("\n  Wähle:")
    if (antwort == "1"):
        VT52_cls_home()
        print("Grad --> Bogenmass")
        gradValue=float(input("Grad:"))
        print("Grad={grad:1.2f} ==> Rad={rad:1.2f}".format(grad=gradValue,rad=grad2Rad(gradValue)))
        halt()

```

Own functions with parameters

+ Easier maintenance

+ Elimination of redundant code


```
from waltisLibrary import *
```

```
doLoop = True
while doLoop:
    VT52_cls_home()
    print("  Umrechnungen")
    print("  =====")
    print("  1: Grad in Bogenmass") # rad = grad*pi/180
    print("  2: Bogenmass in Grad") # grad = rad*180/pi
    print()
    print("  3: Fahrenheit in Celsius") #32F -> 0°C
    print("  4: Celsius in Fahrenheit") #32F -> 0°C
    print()
    print("  0: Schluss")

    antwort = input("\n Wähle:")
    if (antwort == "1"):
        VT52_cls_home()
        print("Grad --> Bogenmass")
        gradValue=float(input("Grad:"))
        print("Grad={grad:1.2f} ==> Rad={rad:1.2f}".format(grad=gradValue,rad=grad2Rad(gradValue)))
        halt()

    if (antwort == "2"):
        VT52_cls_home()
        print("Bogenmass --> Grad")
        radValue=float(input("Rad:"))
        print("Rad={rad:1.2f} ==> Grad={grad:1.2f}".format(rad=radValue,grad=rad2Grad(radValue)))
        halt()

    if (antwort == "3"):
        VT52_cls_home() # http://www.metric-conversions.org/de/temperatur/fahrenheit-in-celsius.htm
        print("Fahrenheit in Celsius")
        fahrenheitValue=float(input("Fahrenheit:"))
```

Function Libraries

- + Easier Maintenance
- + Reuse possible
- + Elimination of redundant code
- + Separation application / reusable libs

```
import math
import os
import sys
import time
import datetime

from time import sleep

# Primzahlen Functions
# -----
def isPrimzahl(aZahl):
    isPrim = True
    if (aZahl == 1):
        isPrim = True
    else:
        if (aZahl == 2):
            isPrim = True
        else:
            obergrenze = int((aZahl / 2) + 2)
            for i in range(2, obergrenze):
                if ((aZahl % i) == 0):
                    isPrim = False
    return isPrim

def getNextPrimzahl(zahl):
    aZahl = zahl + 1
    while (isPrimzahl(aZahl) == False):
        aZahl = aZahl + 1
    return aZahl
```

External Function Libraries

- + Reuse possible
- + Because of Internet
 - + open source
 - + sharing for free
- + IDE simplifies to import
- No states
- Language dependency


```

if (inTestMode):
    print("IN TEST-MODE")
    print("    time_T2                : ",time_T2)
    print("    schwellwert_H_OffImmediate: ",schwellwert_H_OffImmediate)
    print()
    print("=====\\n")
print(hsrKaelteMaschine.toString())
logFile.addLogEntry(hsrKaelteMaschine.toStringForLog())
if (hsrKaelteMaschine.isEmergencyOff_Active()):
    print("-----1")
    print("Emergency Off pressed!! Will not start!!!")
    hsrKaelteMaschine.doEmergencyStop()
    verdichter_1.doEmergencyStop()
    verdichter_2.doEmergencyStop()
    # t2State.setState_NotStarted() # Waterpump steys on in EmercencyOff
else:
    print("-----2")
    vorhandeneEnergie      = hsrKaelteMaschine.getExistingEnergy()
    hochdruck              = hsrKaelteMaschine.getHighPressure()
    niederdruck            = hsrKaelteMaschine.getLowPressure()
    waterTemp              = hsrKaelteMaschine.getWaterTemp_PT1000()
    envTemp                 = hsrKaelteMaschine.getEnvironmentTemp_PT1000()

    # -----
    # Calculated values
    # -----
    lowPressureError        = niederdruckOff.setState(niederdruck,verbal = True)
    highPressureError        = hochdruckOff.setState(hochdruck ,verbal = True)
    pressureError = (lowPressureError or highPressureError)
    if (pressureError):
        if (t1State.isState_NotStarted()):
            t1 = Timer(time_T1, T1Over, args=[t1State])
            t1State.setState_Ticking()
            t1.start()
        print("pressureError      :",pressureError,end="")
        if (lowPressureError):
            print("    (lowPressureError)",end="")
            hsrKaelteMaschine.setShutOffValveToOpen(True)
            time.sleep(lowPressureErrorShutOffValveReopenTime)

```

Classes: Methods with states

- + Reuse improved (inheritance)
- + Decoppeling
- + Simple interfaces
- Language dependend
- No runtime distribution

```

# -----
import math
import time
import datetime

from waltisLibrary import *

class Verdichter:

    # Ctr (Konstruktor)
    # -----
    def __init__(self, name, pMin, pMax, outputTiefpass, hsrKaelteMaschine):
        self.name = name
        self.pMin = pMin
        self.pMax = pMax
        self.outputTiefpass = outputTiefpass
        self.hsrKaelteMaschine = hsrKaelteMaschine
        self.istSpeed = 0
        self.sollSpeed = 0
        self.verdichterOn = False
        self.isVerdichterOnState = False
        self.wiederEinschaltenMoeglich = False
        self.lastTimeOff = datetime.datetime.now()

    # Methoden (setter / getter)
    # -----
    def setValues(self, sollSpeed, emergencyOff, einschaltverzoeigert):
        currTime = datetime.datetime.now()
        wiedereinschaltverzoeigerung = "0:00:10"
        print("==> setValues ({x1:1s})::sollSpeed:{x2:4.0f}%".format(x1=self.name, x2=sollSpeed))

    def (emergencyOff):
        self.sollSpeed = 0
        self.verdichterOn = False
        self.resetRestartTime()
    else:
        if (sollSpeed > self.pMax):
            self.sollSpeed = self.pMax
            self.verdichterOn = True
        elif (sollSpeed < self.pMin):
            self.sollSpeed = 0
            if (self.verdichterOn):
                self.verdichterOn = False

```

Class-Libraries (Packages)

- + Reuse improved
- + Decoupling
- + Simple interfaces
- Language dependend
- No runtime distribution

```
package com.rothlin.hwz;

import java.io.IOException;

@WebServlet("/HWZ_SimpleEJB_Servlet_1")
public class HWZ_SimpleEJB_Servlet_1 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    private TestHWZ_EJB testEJB;

    public HWZ_SimpleEJB_Servlet_1() {
        super();
    }

    private void serviceRequest(
        HttpServletRequest request,
        HttpServletResponse response, boolean callFrom_doGet)
        throws ServletException, IOException {
        StringBuffer sb = new StringBuffer();
        sb.append("<html>\n<head>\n</head>\n<body>\n");
        sb.append("Hallo!!!!" + testEJB.sayHello());
        sb.append("</body>\n</html>\n");
        response.setContentType("text/html");
        response.getWriter().print(sb);
    }
}
```

Distributed Objects

- + Component reuse
- + Scallable
- + Simple interfaces
- Language dependend

```
package com.rothlin.hwz;  
import javax.ejb.LocalBean;
```

```
@Stateless  
@LocalBean
```

```
public class TestHWZ_EJB {
```

```
    public TestHWZ_EJB() {}
```

```
    public String sayHello() {  
        return "Hallo, wo gibt es ein Bier?";  
    }
```

```
}
```

Distributed Objects

+ POJO

- Language dependend

```
private FirstWebServiceProxy ws;

public WebServiceClient00() {
    ws = new FirstWebServiceProxy();
}

public static void main(String[] args) {
    try {
        WebServiceClient00 client = new WebServiceClient00();
        client.doTest(args);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void doTest(String[] args) {
    try {
        System.out.println(ws.ping());
        System.out.println(ws.sayHelloName("BWI-15"));
        System.out.println(ws.sayAnredeName("Herr", "Peter"));
        System.out.println(ws.calcCircleArea(5.0));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Web-Services

+ Simple

+ Simple

+ Language independent

```
@WebService
public class FirstWebService {

    // Methodes in a web-Service are called Operations
    // Operation names can not be overloaded!!
    @WebMethod(exclude=false)
    public String ping() {
        return "Webservice (1.0.0.0)::FirstWebService:ping() called!!!!!!";
    }

    @WebMethod() // ctrl-space shows you the options
    public String sayHelloName(String name) {
        return "Hello " + name;
    }

    @WebMethod() // ctrl-space shows you the options
    public String sayHelloHWZ(String name) {
        return "Hello du " + name + "! Gruss von der HWZ";
    }

    @WebMethod(exclude=false)
```

Web-Services

- + Simple
- + Simple
- + Language independent

Application / Web-Application

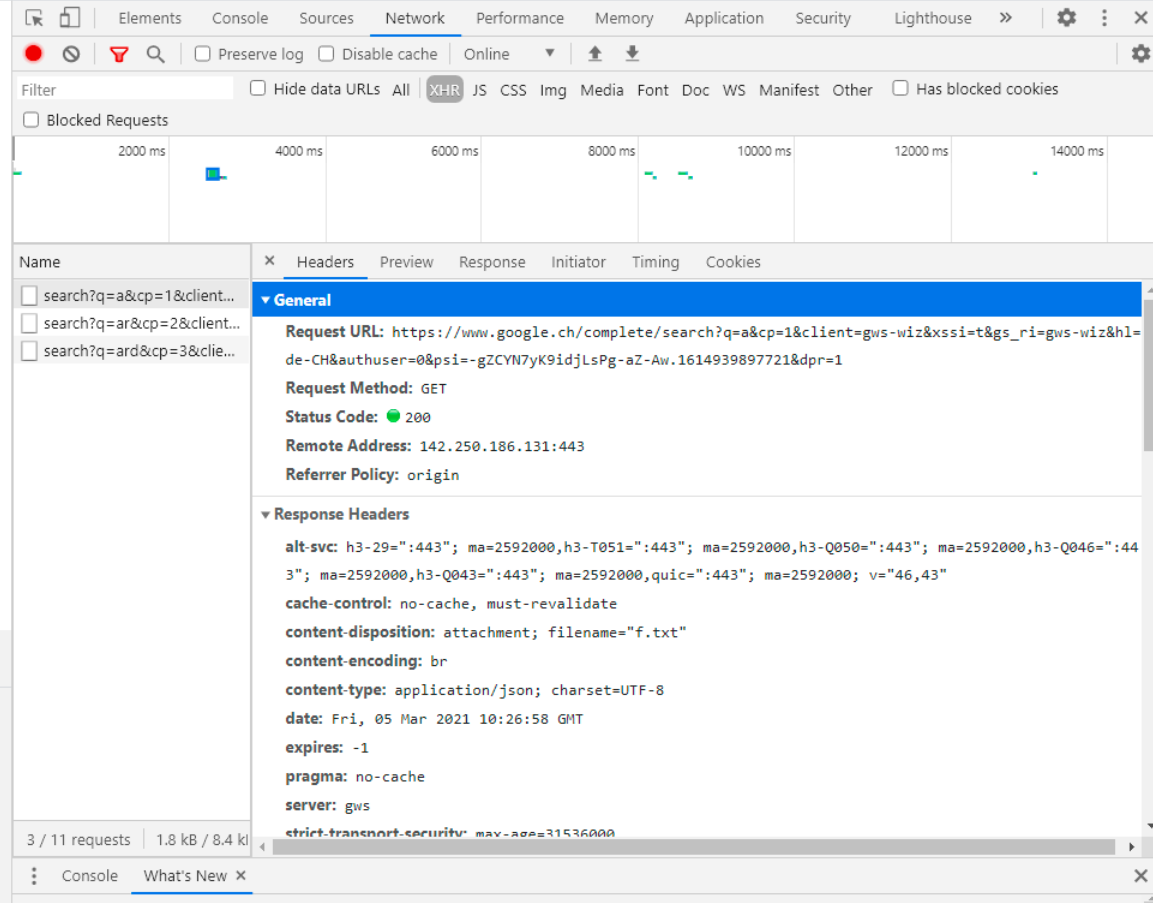
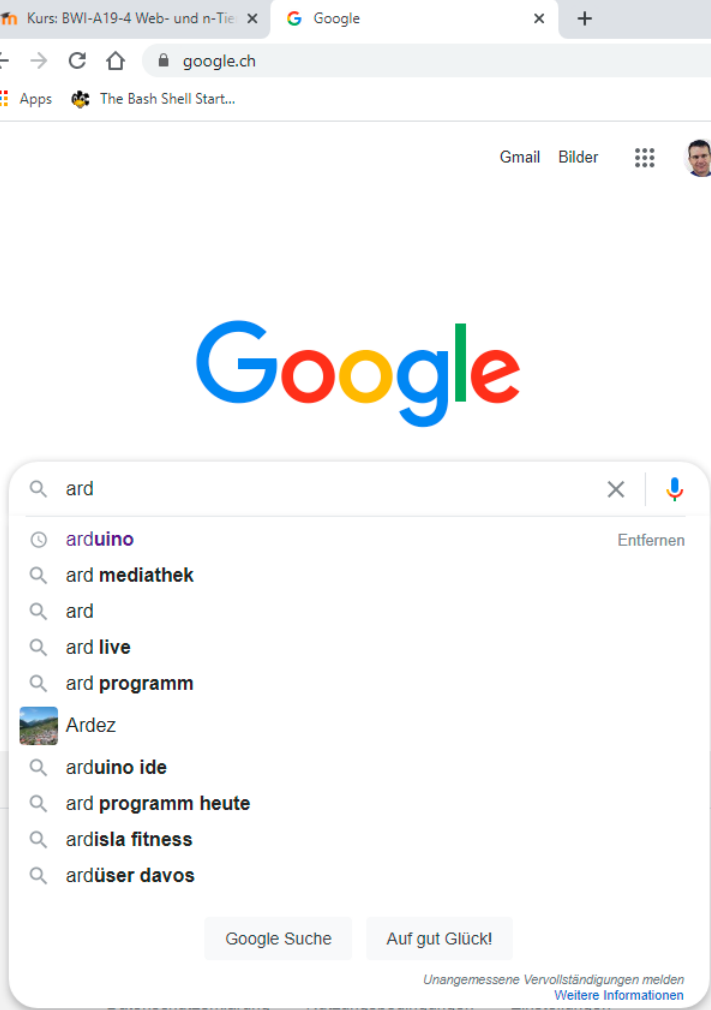
- Process (Programm), der auf Umgebung reagiert
 - User-Interactions
 - Via sensors (input) and actors (output)
- Web / HTML → strukturierte und verlinkte Dokumente
 - Via URL
 - Page based (reload)
 - Client (Web-Browser) sendet requests an den Server (Web-Server)
 - Web-Server sendet eine Response zurück
- Web-Application läuft im Browser
 - Eingabe Felder (Form-Elemente) / Layout (CSS)
 - Abhängig von Events müssen Actions ausgelöst werden → Programmlogik durch JavaScript (JS) im HTML-Dokument
 - Actions können durch JS dynamisch (ohne reload) DOM verändern
 - Actions können auch Request auslösen und die Response wird von JS im DOM verarbeitet

Examples Web-Application

- Arduino: <http://192.168.1.20/>
- SBB Fahrplan: <https://www.sbb.ch/>
- Google search: <https://www.google.ch/>
- URL verkürzen (Bitly): <https://bitly.com/>
- QR-Code generator: <https://www.qrcode-generator.de>
 - Web-Service: <http://goqr.me/de/api/doc/>
 - Beispiel: <http://api.qrserver.com/v1/create-qr-code/?data=HelloWorld!&size=100x100>
- Map Geo: <https://map.geo.admin.ch>
 - API: <http://api.geo.admin.ch/>
 - REST: <https://api3.geo.admin.ch/1912100956/rest/services/ech/SearchServer?sr=2056&searchText=peterliwiese%2033&lang=en&type=locations>
 - Application: <http://www.fhoch3.peterliwiese.ch/AdresseToKoordinates.html>
 - Excel calls REST: <https://www.geo.admin.ch/de/news/aktuell.detail.news.html/geo-internet/news2019/news20190307.html>
 - Konsultieren Sie bitte hierzu api.geo.admin.ch Beispiel zur Nutzung des RESTService zb in dieser excel formel
<https://www.geo.admin.ch/de/news/aktuell.detail.news.html/geo-internet/news2019/news20190307.html>
 - Das API von geo.admin.ch darf auf Webseiten eingesetzt werden, sofern diese Verwendung den Nutzungsbedingungen von GeoAdmin und CadastralWebMap entsprechen. Die Registrierung und Nutzungsbedingungen finden Sie unter:
<https://www.geo.admin.ch/de/geo-dienstleistungen/geodienste/darstellungsdienste-webmapping-webgis-anwendungen/programmierschnittstelle-api/anmeldung.html>
- Open Weather: <https://openweathermap.org/>
 - API: <https://openweathermap.org/api>
 - REST: <http://api.openweathermap.org/data/2.5/weather?q=Wangen%20SZ&appid=3836093dde650898eb014e6f27304646>

Google-Chrome as Web-Dev Tool

Weitere Tools → Entwicklertools



URL Uniform Resource Locator

Eine URL ist eine weltweit eindeutige Adresse er Resource (z.B. Datei)

z.B. <http://www.fh-hwz.ch:80/Download/b1a/Semesterprogramm.pdf>

http:	Protokoll (https)	Wie?
www.fhhwz.ch:	Full qualified host name	Mit wem?
80:	Port-Nummer	
Download/b1a:	Directory-Struktur	Was?
Semesterprogramm.pdf:	Document name	

Definiert wie zwei Kommunikations-Partner miteinander sprechen.

Im World Wide Web (www)

http Hyper text transfer protokoll

https http-Secure

Weitere Internet Protokolle

ping Verbindung testen

telnet remote Console

ftp File transfer

sftp secure file transfer (Public-Private Key)

ftps secure file transfer (Certificates)

smtp simple mail tranfer protokoll

Host- und Domain-Names

Jeder Computer in einem Netzwerk muss eindeutig bezeichnet sein!

→ IP Address:

↳ Eindeutige Nummer (4-stellig, jede Stelle 0..255: $255^4=4\text{Mia}$)

↳ z.B. 107.18.128.198

Nummern kann man sich schlecht merken! Namen bringen mehr Flexibilität!

→ Hostname / Domain-Name:

↳ Eindeutiger Name für einen Computer

↳ z.B. celera.credit-suisse.ch

2nd Level Domain

1st Level Domain

Domain-Name
Hostname = Full qualified hostname

Übersetzt IP-Adressen in Hostnames und umgekehrt: → DNS (Domain Name Service)

`ipconfig /all`

zeigt die eigene IP Adresse

`nslookup`

übersetzt IP ← → Hostname

Port

Auf einem Computer sind mehrere Zuhörer / Sprecher gleichzeitig aktiv. Der Port bestimmt mit welchem ich sprechen will.

Dienste	Port	Protokoll
echo	7	TCP
systat	11	TCP
chargen	19	TCP
ftp-data	21	TCP
ssh	22	TCP
telnet	23	TCP
smtp	25	TCP
nameserver	42	TCP
whois	43	TCP
tacacs	49	UDP
dns-lookup	53	UDP
dns-zone	53	TCP
orac1-sqlnet	66	TCP
tftp	69	UDP
finger	79	TCP
http	80	TCP
http-1	81	TCP
kerberos	88	TCP
pop2	109	TCP
pop3	110	TCP
sunrpc	111	TCP
sqlserv	118	TCP

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

0 to 1023

are the *well-known ports* or *system ports*

1024 to 49151

are the registered ports

e.g. 8080 alternate http

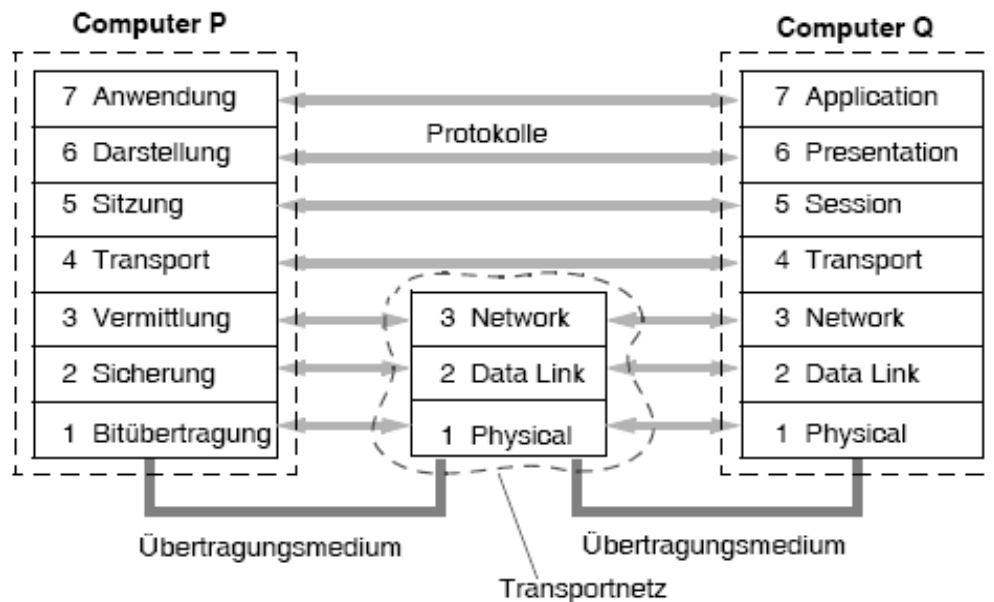
49152–65535 ($2^{15}+2^{14}$ to $2^{16}-1$)

contains dynamic or private ports

TCP/IP Packet

IP Header	Version	IHL	Type of Service		Total Length		32-Bit / Zeile		
	Identification			Flags	Fragment Offset				
	Time to Live		Protocol=6 (TCP)		Header Checksum				
	Source Address							IP-Adresse z.B. 158.2.34.150	
	Destination Address								
Options					Padding				
TCP	Source Port			Destination Port			Port Nr: 0.. 65535		
	Sequence Number								
	Acknowledgement Number								
	Data Offset		U R G	A C K	P S H	R S T		F I N	Window
	Checksum			Urgent Pointer					
	TCP Options					Padding			
	TCP Data								

TCP/IP Open Systems Interconnect (OSI) Reference Model



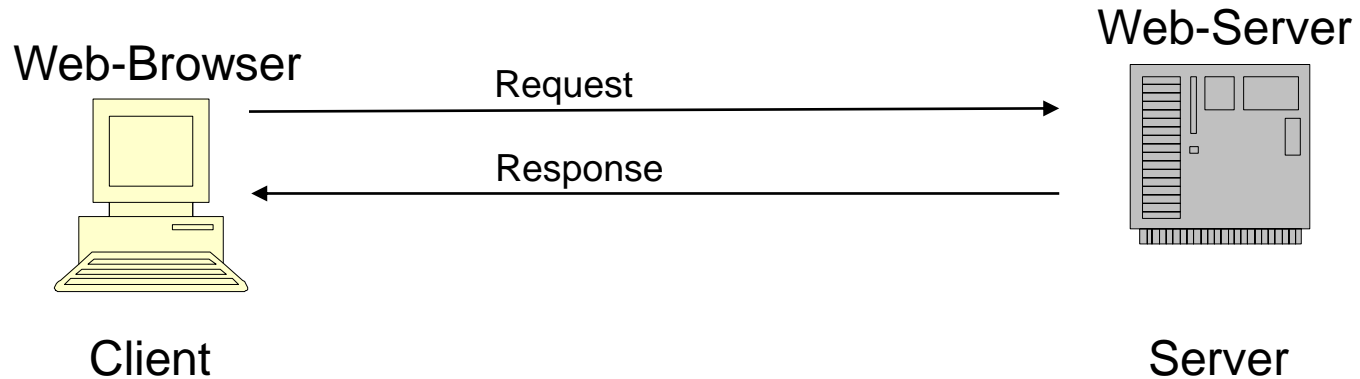
7	Anwendungsschicht besteht aus den Anwendungen mit denen man das Netz nutzen kann
6	Darstellungsschicht standardisiert das Format der Daten auf dem Netz
5	Kommunikationssteuerungsschicht verwaltet die Verbindungen zwischen den Anwendungen
4	Transportschicht garantiert die fehlerfreie Datenübertragung durch Fehlererkennung und -korrektur
3	Vermittlungsschicht verwaltet die Verbindungen zwischen den Rechnern im Netz für die höheren Schichten
2	Sicherungsschicht sorgt für die zuverlässige Übertragung der Daten über die physikalischen Verbindungen
1	Bitübertragungsschicht definiert die physikalischen Eigenschaften der Übertragungswege

Web-Server

Auf einem Server (Computer mit IP Adresse eindeutig in einem Netzwerk bestimmt) ist auf einem (Software-)Port (Default 80) ein Prozess am laufen, welcher http versteht → **Web-Server**

- **Document-Root**: Legt oberstes Verzeichnis des Web-Servers im file-System fest
- **Mime-Type** Table: Zuordnung von File-Extensions zu Mime-Types. Wird vor dem zurück-streamen des Content (File-Inhalt der URL)
- Unterscheidungsmerkmal zwischen **static** und **dynamic-Content**
- Interpretiert URL in http **request** und generiert eine **respond**
- Sobald die Antwort gegeben ist, vergisst der Web-Server alles über den Request (**stateless**)

Zusammenfassung:



1. Web-Server muss laufen; Web-Browser starten
2. **URL** im Browser eintippen und mit Return abschliessen
3. **Request** wird abgesendet und sucht sich den Weg zum Web-Server (via Internet, IP-Adresse und Port)
4. Web-Server interpretiert Request und gibt eine **Response** (mit Mime-Type) zurück
5. Browser **rendert** Antwort

Static Web Pages: HTML (Hypertext Markup Language)

- Text-Datei! Kann mit normalem Text-Editor erstellt werden (z.B. Notepad)
- Darstellung (Layout) wird vom Empfänger definiert (Rendern) und ist somit für Publisher undefiniert.
- Definiert mit Hilfe von Tags die Struktur des Dokuments
- <http://www.peterliwiese.ch/AppletLib/tips/tips.html>

Hello World Beispiel

`<HTML>`

Start-Tag

`<HEAD>`

`<TITLE>First example</TITLE>`

`</HEAD>`

`<BODY bgcolor="#FFFFFF">`

`<H1>Hallo Besucher!</H1>`

Attribute

Value

Dein Besuch freut mich

`</BODY>`

`</HTML>`

End-Tag

Hilfe und Erklärungen auf <http://de.selfhtml.org/>

Cascaded Style Sheet (CSS)

HTML ist eine Markup-Language.

➔ Definiert nur Struktur des Dokuments (nicht Layout)

➔ Mit CSS können Elemente genau positioniert oder formatiert werden

Einbinden von CSS

```
<head>

<title>Das style-Attribut</title>

<link rel="stylesheet"                href="formate.css">
<link rel="stylesheet" media="screen"  href="website.css">
<link rel="stylesheet" media="print, embossed" href="druck.css">
<link rel="stylesheet" media="aural"    href="speaker.css">

<style type="text/css">

/* ... hier sind dateispezifische Formate erlaubt ... */

</style>

</head>
```

http://de.wikipedia.org/wiki/Cascading_Style_Sheets

<http://de.selfhtml.org/css/index.htm>

Cascaded Style Sheet (CSS)

Beispiel eines CSS für HTML-Elemente

```
<style type="text/css">
```

```
body { background-color:#FFFFCC; margin-left:100px; }
```

```
h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin black; }
```

Gilt fuer diese Elemente. Unterelemente

```
* { color:blue; }
```

Universalselektor; gilt fuer alle Elemente

```
p,li { font-size:110%; line-height:140%; font-family:Helvetica,Arial,sans-serif; word-spacing:0.3em; }
```

Eigenschaften gelten für <p> und

```
h1 i { color:blue; font-style:normal; }
```

Eigenschaften gelten nur für <i> innerhalb von <h1>

```
div * b { color:violet; }
```

Eigenschaften gelten nur für innerhalb (beliebig tief) von <div>

```
</style>
```

Cascaded Style Sheet (CSS)

Beispiel eines CSS mit attributbedingten Formaten

```
<style type="text/css">
```

```
p[align] { color:red; }
```

Gilt für `<p>` bei denen das Attribut **align** gesetzt ist

```
p[align=center] { color:blue; text-align:left; }
```

Gilt für `<p>` bei denen das Attribut **align** auf **center** gesetzt ist

```
td[abbr~=Berlin] { background-color:#FFFF00 }
```

Gilt für `<td>` bei denen das Attribut **abbr** den Text **Berlin** enthaelt

```
*[lang|=en] { background-color:#FF0000; color:#FFFFFF; }
```

Gilt für alle Elemente bei denen das Attribut **lang** mit dem Text **en** beginnen

```
</style>
```

Cascaded Style Sheet (CSS)

Beispiel eines CSS und deren Anwendung (class)

```
<style type="text/css">
  .beitrag { border:1px outset gray; margin:.5em; padding:.5em; background-color:#efd;}
  .uebersetzung { border-bottom:1px dotted #900; }
  .autor { font-style:italic; }
  .datum { font-size:80%; color:#444; }
  h1.hinterlegt { background-color:#FFFF00 }
</style>
```

Unterschied <div> und :

- Beide tags relativ eigenschaftslos
- <div> Neue Zeile
- kein neuer Absatz (im Textfluss)

```
<div class="beitrag">
  <p>Beispielsweise in der
    <a href="http://www.w3.org/TR/REC-html40/">HTML 4.01-
      <span lang="en" title="Empfehlung" class="uebersetzung">Recommendation</span></a>.
  </p>
  <p class="autor">Max Hilfreich, <span class="datum">22.05.2006</span> </p>
  <h1 class="hinterlegt">H1 knallgelb hinterlegt</h1>
</div>
```

Cascaded Style Sheet (CSS)

Beispiel eines CSS und deren Anwendung (id)

```
<style type="text/css">
  #roterBereich {
    position:absolute; top:130px; left:30px; width:320px;
    padding:10px; margin:0px; border:4px solid #EE0000;
  }
</style>
```

```
<div id="roterBereich"><h1>Der rote Bereich</h1></div>
```

Layout / CSS: <http://www.dynamicdrive.com/style/>

Using a CSS/JS Library (e.g. Bootstrap)

Build fast, responsive sites with Bootstrap:

<https://getbootstrap.com/>

Themes by Bootstrap's creators:

<https://themes.getbootstrap.com/>

Bootstrap Icons:

<https://icons.getbootstrap.com/>

Design and build your Homepage by using Bootstrap

HTML Form-Elemente

<http://www.peterliwiese.ch/AppletLib/formulare/formularElemente.html>

	<pre><FORM METHOD=POST ACTION=http://www.peterliwiese.ch/cgi-bin/waltisExamples/echo.pl></pre>
RADIO <input type="radio"/> Radio 1 <input checked="" type="radio"/> Radio 2 <input type="radio"/> Radio 3 <input type="radio"/> Radio 4	<pre>RADIO <INPUT TYPE=RADIO NAME=RadioButn VALUE=1>Radio 1 <INPUT TYPE=RADIO NAME=RadioButn VALUE=2 CHECKED>Radio 2 <INPUT TYPE=RADIO NAME=RadioButn VALUE=3>Radio 3 <INPUT TYPE=RADIO NAME=RadioButn VALUE=4>Radio 4</pre>
SELECT (SIZE=1) <input type="text" value="Option 2"/>	<pre>SELECT (SIZE=1) <SELECT NAME=SelectList_1> <OPTION VALUE=1>Option 1 <OPTION VALUE=2 SELECTED>Option 2 <OPTION VALUE=3>Option 3 </SELECT></pre>
SELECT (SIZE=2) <input type="text" value="Option 1"/> <input type="text" value="Option 2"/>	<pre>SELECT (SIZE=2) <SELECT NAME=SelectList_2 SIZE=2 MULTIPLE> <OPTION VALUE=1>Option 1 <OPTION VALUE=2 SELECTED>Option 2 <OPTION VALUE=3>Option 3 <OPTION VALUE=4>Option 4 <OPTION VALUE=5>Option 5 </SELECT></pre>
CHECKBOX <input checked="" type="checkbox"/> Checkbox 1 <input type="checkbox"/> Checkbox 2	<pre>CHECKBOX <INPUT TYPE=CHECKBOX NAME=Checkbox1 CHECKED> Checkbox 1 <INPUT TYPE=CHECKBOX NAME=Checkbox2>Checkbox 2</pre>
TEXT <input type="text" value="Walter_Rothlin"/>	<pre>TEXT <INPUT TYPE=TEXT NAME=TextfieldNormal></pre>
PASSWORD <input type="password" value="•••••"/>	<pre>PASSWORD <INPUT TYPE=PASSWORD NAME=Passwort></pre>

HTML Form-Elemente

<http://www.peterliwiese.ch/AppletLib/formulare/formularElemente.html>

TEXTAREA (with Wrap)

Default Text

TEXTAREA (with Wrap)
<TEXTAREA NAME=Textarea ROWS=3 COLS=50 WRAP=virtual>Default Text</TEXTAREA>

TEXTAREA (with Wrap)

Default Text 1

TEXTAREA (with Wrap)
<TEXTAREA NAME=Textarea1 ROWS=5 COLS=50 WRAP=Off>Default Text 1</TEXTAREA>

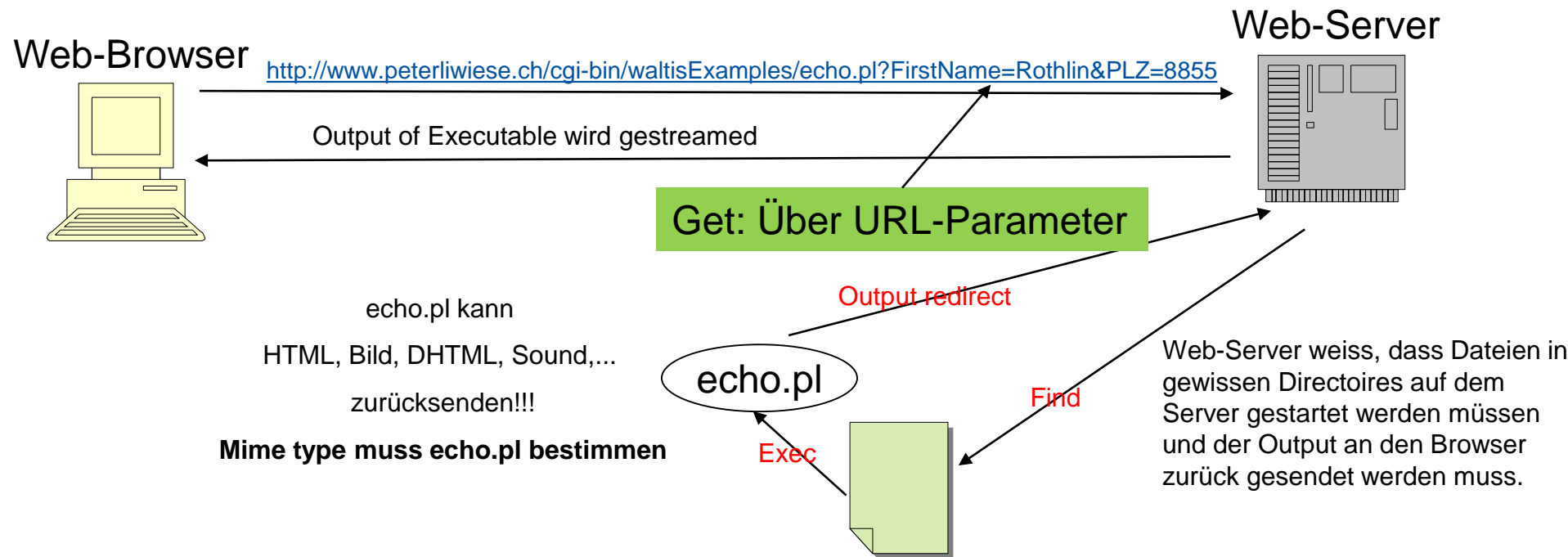
Reset

Submit

<INPUT TYPE=RESET VALUE="Reset">
<INPUT TYPE=SUBMIT NAME=submitted VALUE="Submit">
</FORM>

Vollständige Definitionen: <http://www.w3.org/TR/html401/interact/forms.html>

Dynamic HTTP-Roundtrip



Teile von CGI:

- Code ausführen via URL Request
- Parameter über ENV zur Verfügung stellen
- Output redirect

Nachteil:

Bei jedem Request wird neuer Prozess gestartet (Ressourcen, Stateless)

Unterschied POST und GET

<http://www.peterliwiese.ch/AppletLib/formulare/echo.html>

Name:

Text:

```
<form action="/cgi-bin/waltisExamples/echo.pl" method=post>
  Name: <input size=40 maxlength=40 name="AnwenderName_1" Value="Ein Text mit &Uuml; &uuml; &sauml;">
  Text: <textarea rows=5 cols=30 name="Kommentartext_1" wrap=virtual></textarea>
  <input type=submit value="Absenden mit POST">
</form>
```

Post: Über Header-Parameter

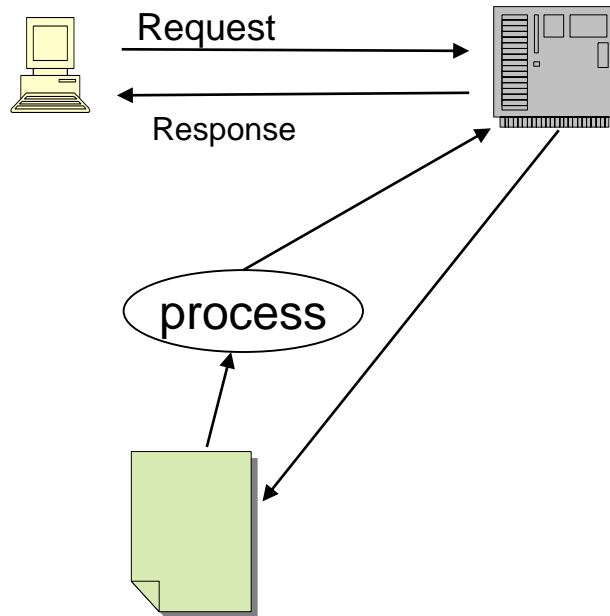
Name:

Text:

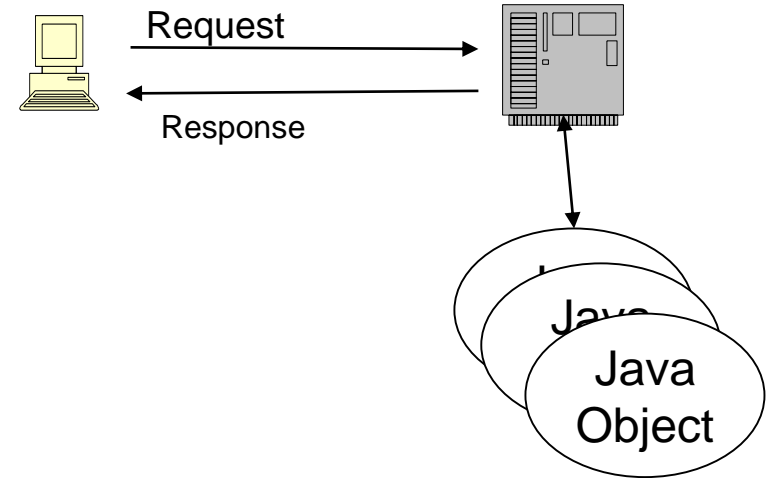
```
<form action="/cgi-bin/waltisExamples/echo.pl" method=get>
  Name: <input size=40 maxlength=40 name="AnwenderName" Value="Ein Text mit &Uuml; &uuml; &sauml;">
  Text: <textarea rows=5 cols=30 name="Kommentartext" wrap=virtual></textarea>
  <input type=submit value="Absenden mit GET">
</form>
```

Get: Über URL-Parameter

Servlet: Cgi versus Java way



- Parameter handling
- Return String
- Process creation
- Not object oriented
- No sessions and states



- + Objects already created
- + Nice interface
- + Proper Parameter handling in / out
- + Can handle states and sessions

Java-Script (einbinden)

```
// extern JS file
<script src="JS_Library.js" type="text/javascript"></script>

// code in Header
<script language="JavaScript">
<!-- Hide Script from older Browsers.
function saveValuesInCookies(form) {
    setCookie('Name',form.Name.value)
    setCookie('Vorname',form.Vorname.value)
    setCookie('Wohnort',form.Wohnort.value)
    setCookie('PLZ',form.PLZ.value)
    return;
}
//-->
</SCRIPT>
```


Java-Script (Event-Handlers)

Aus einem HTML-Link

```
<A HREF="javascript:saveValuesInCookies (document.adressForm) ;">Save Values </A>
```

Event-Handler auf Formelement

<http://de.selfhtml.org/javascript/sprache/eventhandler.htm>

```
<input type="button" onclick="checkBeforeSubmit (document.adrForm) " value="Submit">  
<textarea rows="5" cols="40" onchange="alert (this.value)">Ein Text<textarea>
```

Event-Handler im BODY-TAG

```
<body onload="getValuesFromCookies (document.adressForm) ">
```

Java-Script (code)

```
// function to check form content before sending to server
// -----
function checkBeforeSubmit (form) {
    if (!check(form)) return;
    form.submit();
    return;
}

function check(form) {
    form.Name.value = form.Name.value.toUpperCase();
    if (form.Name.value.length != 7 ) {
        alert ("Name has to be exactly 7 char long (" + form.Name.value + ")");
        form.Name.focus();
        return (false);
    }

    if ((form.PLZ.value.length != 4 ) || (isNaN(parseInt(form.PLZ.value)))) {
        alert ("PLZ must be a 4 digit number");
        form.PLZ.focus();
        return (false);
    }

    saveValuesInCookies(form)

    return (true);
}
```

Simulate stateful sessions: Cookies

Textinformation, die vom Server (im Header oder via Client-Script) an den Client übertragen wird.

Beim nächsten http-Request werden alle Cookies dieser Seite mit dem Request mitgesendet.

➔ kann nun vom Server ausgewertet werden (z.B. Session-Id, welche dann zum Warenkorb führt)

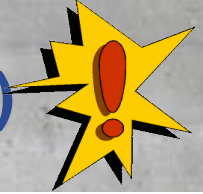
"Set-Cookie:" Name "=" Wert *("; " Attribut)

Name=Wert ist eine Folge von encoded Zeichen. z.B. SessionId=12345678

Folgende Attribute sind in der Spezifikation von Netscape definiert:

- **EXPIRES=***dateValue*
Nach diesem Datum / Zeit wird Cookie gelöscht und nicht mehr gesendet
- **DOMAIN=***domainName* (Default current Domain)
Cookie wird nur mitgesendet bei Requests an diesen Domain-Name und deer Sub-Domains
- **PATH=***pathName* (Default current Path)
Cookie wird nur mitgesendet bei Requests an diesen Path
- **SECURE**
Cookie wird nur gesendet, falls der Request über https geht

Simulate stateful sessions: Cookies (via JavaScript)



```
function setCookie(name, value) {  
    var today = new Date();  
    var expire = new Date();  
    expire.setTime(today.getTime() + 1000*60*60*24*365); // Cookie lebt 1 Jahr  
    document.cookie = name + "=" + escape(value) + ";expires=" + expire.toGMTString();  
}
```

Setzen:

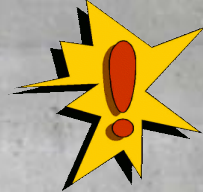
```
setCookie('emailAdr', form.emailAdr.value)
```

```
function getCookie(name,defaultValue) {  
    var search = name + "=";  
    if (document.cookie.length > 0) {  
        offset = document.cookie.indexOf(search);  
        if (offset != -1) {  
            offset += search.length  
            end = document.cookie.indexOf(";", offset);  
            if (end == -1)  
                end = document.cookie.length;  
            return unescape(document.cookie.substring(offset, end));  
        } else {  
            return defaultValue;  
        }  
    } else {  
        return defaultValue;  
    }  
}
```

Lesen:

```
form.emailAdr.value = getCookie('emailAdr',")
```

Simulate stateful sessions: Hidden Parameter



Dieses Beispiel: <http://www.peterliwiese.ch/FormHandler/testFormBW.html>

```
<BR>
e-mail Adresse
<INPUT TYPE=TEXT NAME=emailAddr VALUE="Walter@Rothlin.com">
<INPUT TYPE=HIDDEN NAME=constrain_emailAddr VALUE="isEmailAddr(XXXX,Keine g&#252;ltige e-mail Adresse)">
<BR>

<HR>
Kommentar (with Wrap)<BR>
<TEXTAREA NAME=Kommentar ROWS=10 COLS=70 WRAP=virtual>Default Text</TEXTAREA>
<BR>

<HR>
Kommentar 2 (no Wrap)<BR>
<TEXTAREA NAME=Kommentar_2 ROWS=5 COLS=50 WRAP=Off>Default Text</TEXTAREA>

<BR>
<HR>

<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_replyTemplateSuccess VALUE="http://www.peterliwiese.ch/FormHandler/testSuccessTemplate.html">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_replyTemplateNotSuccess VALUE="http://www.peterliwiese.ch/FormHandler/testNotSuccessTemplate.html">

<!--
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_replyTemplateSuccess VALUE="testSuccess.html">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_replyTemplateNotSuccess VALUE="testNotSuccess.html">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailTemplate VALUE="test.txt">
-->
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailTemplate VALUE="http://www.peterliwiese.ch/FormHandler/testEmailTemplate.html">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailSubject VALUE="Antwort (Default as HTML)">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailToAdrrs VALUE="Walter@Rothlin.com">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailCcAdrrs VALUE="">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailBccAdrrs VALUE="">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailFromAddr VALUE="FormField:emailAddr">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailSentAsHTML VALUE="YES">

<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailTemplate_1 VALUE="http://www.peterliwiese.ch/FormHandler/testEmailTemplate.txt">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailSubject_1 VALUE="Antwort as Normal">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailFromAddr_1 VALUE="Walter@Rothlin.com">
<INPUT TYPE=HIDDEN NAME=cgiCmdHidden_emailSentAsHTML_1 VALUE="NO">
```

Beispiel «Ausgefüllter Einzahlungsschein»

Test-Bestellformular Tickets

Anzahl Tickets à CHF 45.-: 2

Lieferung: ☒ Per eingeschriebener Post (+ CHF 6.00)
☐ Abholen bei Daniel Vogt, 8855 Wangen

Rechnungsbetrag (CHF): 96.00

Ihre Angaben

Name: ROTHLIN

Vorname: Walter

Strasse Nr.: Peterliwiese 33

PLZ: 8855 Ort: Wangen

email: walter@rothlin.com

Kommentar

Test (sollte nicht bei Simon ankommen, sonst bitte sofort melden)

Kostenpflichtig bestellen

Rechnung Ticket-Bestellung Oktoberfest Wangen SZ

Rechnungs-Nummer: 001029 (Bitte bei Zahlung angeben)

2 Ticket(s) zu CHF 45.--

Lieferung: Wird nach Eingang der Zahlung per Eingeschriebener-Post gesendet plus CHF 6.00

Zu bezahlender Betrag: -----96.00 CHF

Ihre Kommentar:

Test (sollte nicht bei Simon ankommen, sonst bitte sofort melden)

Empfangsschein / Récépissé / Ricevuta	Einzahlung Giro	Versement Virement	Versamento Girata
<p>Einzahlung für / Versement pour / Versamento per Schwyzer Kantonalbank 6431 Schwyz</p> <p>Zugunsten von / En faveur de / A favore di CH41 0077 7005 8249 1145 5 Feuerwehrverein Wangen-Nuolen 8855 Wangen SZ</p> <p>Konto / Compte / Conto CHF 60-1-5</p> <p>Einbezahl von / Versé par / Versato da Walter ROTHLIN Peterliwiese 33 8855 Wangen</p>	<p>Einzahlung für / Versement pour / Versamento per Schwyzer Kantonalbank 6431 Schwyz</p> <p>Zugunsten von / En faveur de / A favore di CH41 0077 7005 8249 1145 5 Feuerwehrverein Wangen-Nuolen 8855 Wangen SZ</p> <p>Konto / Compte / Conto CHF 60-1-5</p> <p>Einbezahl von / Versé par / Versato da Walter ROTHLIN Peterliwiese 33 8855 Wangen</p>	<p>Zahlungszweck / Motif versement / Motivo versamento Oktoberfest: 001029</p>	
<p>CHF 96.00</p>			
<p>303</p>			
<p>Die Annahmestelle L'office de dépôt L'ufficio d'accettazione</p>			
<p>Vielen Dank für Ihre Bestellung</p>			

<http://www.peterliwiese.ch/oktoberfest/TestBestellformular.shtml> (Beispiel zum Ausprobieren)

<http://www.hwz.brunnenhoefli.ch/uebungen.html> (Beispiel 1.6 für Source-Code)

Übung Formularhandler



www.peterliwiese.ch

Web-Server

Formhandler

email

hwz.peterliwiese.ch

Web-Server

HTML
Formular

Templates

Client-Browser:

Microsoft Internet Explorer

http://www.peterliwiese.ch/formHandler.pl

**Daten wurden an
Walter@Rothlin.com
übermittelt**

Vielen Dank Herr Corsini für Deine Angaben

Passwort war Ein Passwort

Vielen Dank

Microsoft Internet Explorer

Kundenformular

http://www.hetzl.ch/.../formHandler.html

**Formularwerte werden zu formHandler.pl
auf einem anderen Server gesendet**

Name: Rothlin
Vorname: Walt
email: Walter@Rothlin.com
PLZ: Ort:
Reset Submit